# MỘT SỐ HƯỚNG DẪN LẬP TRÌNH C

## I. MỘT SỐ NGUYÊN TẮC TRÌNH BÀY CHƯƠNG TRÌNH NGUỒN

### 1. Cách đặt tên hàm, biến, kiểu dữ liệu, hằng

- Khi tên hàm, tên biến là sự kết hợp của nhiều từ thì các từ được viết liền nhau (không nên dùng dấu gạch dưới _ để phân cách), kí tự đầu của mỗi từ viết hoa, các kí tự còn lại của mỗi từ viết thường.
  - Ví dụ : **LoadData, FindShortestWay, NumLines, GraphData, …**
- Tên biến : thường bắt đầu bằng một danh từ. Tuy nhiên để thuận tiện cho việc theo dõi chương trình, người ta thường thêm trước tên biến một số kí tự viết thường để chỉ kiểu dữ liệu của biến đó. Các kí tự thường được dùng trong qui ước này thường là :

| | | |
|---|---|---|
| **sz** : kiểu chuỗi | **c** : kiểu kí tự | **f** : kiểu số thực |
| **n** : kiểu số nguyên | **p, lp** : kiểu con trỏ | **l** : kiểu số long |

  - Ví dụ :
    **char** *szFileName; // **szFileName** là biến kiểu chuỗi
    **LPEVENT** lpEvent; // **lpEvent** là biến kiểu con trỏ
- Tên hàm : thường bắt đầu bằng một động từ. Thứ tự các tham số trong hàm được qui ước theo thứ tự : các dữ liệu trả về, các dữ liệu vào, …
  - Ví dụ :
    **void CopyArray(int** *Dest**, int** *Src**, int** NumElement**);**
    **int LoadData(char** *szFName**);**
- Tên hằng, kiểu dữ liệu thường được viết bằng chữ in hoa.
  - Ví dụ :

```
#define MAX 10
#define TIMEDELAY  50
typedef struct {
    int Left, Top;
    int Right, Bottom;
}RECT;
```

### 2. Cách trình bày

- Đầu mỗi chương trình hay tập tin đều có một số dòng mô tả. Các thông tin thường đề cập trong phần này thường là : tên tập tin, tóm tắt mục đích của chương trình, môi trường làm việc, trình biên dịch, tác giả, ngày cập nhật cuối cùng, …
  - Ví dụ :

```
/*************************************************************************************************
*   File Name      : GRAPH.CPP – Graph Function : Load data, display, …                 *
*   Compiler       : BC++3.1.                                                           *
*   Written by     : Le Dinh Duy – Info Dept, HCMUNS.                                   *
*   Last Update    : 03/12/1998                                                         *
*************************************************************************************************/
```

- Mỗi câu lệnh viết trên một dòng. Các câu lệnh cùng cấp viết trên cùng một cột, các câu lệnh có cấp nhỏ hơn viết thụt vào trong, cách lệnh cấp trên bằng một khoảng Tab (thường đặt **Tab Size=4**)

  - Ví dụ :

```
void Demo(void)
{
    int x, y;

    printf("Input x, y : ");
    scanf("%d%d", &x, &y);
    if(x!=y)
        printf("Input error");
    else
        printf("Input valid");
}
```

## II. HƯỚNG DẪN LẬP TRÌNH PROJECT

### 1. Một số thao tác :

- Tạo một project : chọn menu **Project/Open Project**. Tập tin project thường có phần mở rộng là **.PRJ**.
- Hiện cửa sổ project : chọn menu **Window/Project**.
- Thêm/ bỏ một tập tin vào/ra project : mở cửa sổ project, dùng phím **Ins** để thêm, phím **Del** để bỏ.
- Đóng một project : chọn menu **Project/Close Project**.

### 2. Một số nguyên tắc khi kết nối dữ liệu trong Project.

- Các tập tin **.H** thường dùng để khai báo các biến dữ liệu và hàm dùng chung (**export data**). Các thể hiện cụ thể của các biến và hàm này được cài đặt trong tập tin **.CPP** tương ứng.
- Để tránh khai báo trùng lặp khi kết nối các tập tin vào trong project, bắt đầu và kết thúc tập tin **.H** thường thêm một số dòng :

```
#ifndef _ _ <Ten tap tin> _ H _ _  // Ví dụ #ifndef __GLOBAL_H__
#define _ _ <Ten tap tin> _ H _ _

// Cac khai bao ghi o day

#endif // Ket thuc tap tin
```

- Các kiểu dữ liệu, hằng, **macro** dùng chung khai báo trong các tập tin **.H** để khi dùng thì **include** vào.
- Các thành phần dùng chung (biến, hàm) khai báo trong tập tin **.H** bắt đầu bằng từ khóa **extern**, thể hiện của chúng chỉ khai báo một lần duy nhất trong tập tin **.CPP** tương ứng.
- Tập tin project thường chứa các tập tin cài đặt **.CPP**, thư viện đối tượng **.OBJ, ...**
- Các tập tin trong cùng một project thường được tổ chức trong cùng một thư mục để tiện cho việc lưu trữ, kết nối.